Effects of Community Structure on Search and Ranking in Information Networks

Huafeng Xie[1,3], Koon-Kiu Yan[2,3], Sergei Maslov[3] *
[1]*New Media Lab, The Graduate Center,*
CUNY New York, NY 10016, USA
[2]*Department of Physics and Astronomy,*
Stony Brook University,
Stony Brook, New York, 11794, USA
[3]*Department of Physics, Brookhaven National Laboratory,*
Upton, New York 11973, USA

(Dated: February 2, 2008)

The World-Wide Web (WWW) is characterized by a strong community structure in which communities of webpages (e.g. those sharing a common keyword) are densely interconnected by hyperlinks. We study how such network architecture affects the average Google ranking of individual webpages in the comunity. It is shown that the Google rank of community webpages could either increase or decrease with the density of inter-community links depending on the exact balance between average in- and out-degrees in the community. The magnitude of this effect is described by a simple analytical formula and subsequently verified by numerical simulations of random scale-free networks with a desired level of the community structure. A new algorithm allowing for generation of such networks is proposed and studied. The number of inter-community links in such networks is controlled by a temperature-like parameter with the strongest community structure realized in "low-temperature" networks.

PACS numbers: 89.20.Hh, 05.40.Fb, 89.75.Fb

The World Wide Web (WWW) – a very large ($\sim 10^{10}$ nodes) network consisting of webpages connected by hyperlinks – presents a challenge for the efficient information retrieval and ranking. Apart from the contents of webpages, the topology of the network itself can be a rich source of information about their relative importance and relevance to the search query. It is the effective utilization of this topological information [1] which advanced the Google search engine to its present position of the most popular tool on the WWW and a profitable company with a current market capitalization around $30 billion. To rank the importance of webpages Google simulates the behavior of a large number of "random surfers" who just follow a randomly selected hyperlink on each page they visit. The number of hits a given page gets in the course of such simulated process determines its ranking. It is intuitively clear that the larger is the number of hyperlinks pointing to a given webpage (its in-degree in the network) the higher are the chances of a random surfer to click on one of them and, therefore, the higher would be the resulting Google rank of this webpage. However, the algorithm goes beyond just ranking nodes based on their in-degrees. Indeed, the traffic directed to a given webpage along a particular incoming hyperlink is proportional to the popularity of the webpage containing this link. Therefore, the Google rank of a node is given by the weighted in-degree where the weight of each neighboring webpage reflects its importance and is determined self-consistently. The WWW is a very heterogeneous collection of webpages which can be grouped based on their textual contents, language in which they are written, the Internet Service Provider (ISP) where they are hosted, etc. Therefore, it should come as no surprise that the WWW has a strong community structure [2] in which similar pages are more likely to contain hyperlinks to each other than to the outside world. Formally a web community can be defined as a collection of webpages characterized by a higher than average density of links connecting them to each other. In this letter we are going to address the question: how the community structure affects the Google rank of webpages inside the community. One might naively expect that the community structure always boosts the Google rank of its webpages as it tends to "trap" the random surfer inside the community for a longer time. However, it turned out that it is not generally true. In fact the Google rank of community webpages could either increase or decrease with the density of inter-community links depending on the exact balance between average in- and out-degrees in the community. In the heart of the Google search engine lies the PageRank algorithm determining the global "importance" of every web page based on the link structure of the WWW network around it. While the details of the algorithm have undoubtedly changed since its introduction in 1997, the central "random surfer" idea first described in [1] remained essentially the same. To a physicist the algorithm behind the PageRank just simulates an auxiliary diffusion process taking place on the network in question. Similar diffusion algorithms have been recently applied to study citation and metabolic networks [4] and the modularity of the Internet on the "hardware level" represented by an undirected network of interconnections between Autonomous Systems [5]. A large number of

*To whom the correspondence should be addressed: maslov@bnl.gov random walkers are initially randomly distributed on the network and are allowed to move along its directed links. As in principle some nodes in the network could have zero out-degree but non-zero in-degree and would thus "trap" random walkers, the authors of the algorithm introduced a finite probability α for a surfer to randomly select a page in the network and directly jump there without following any hyperlinks. This leaves the probability $1-\alpha$ for a surfer to randomly select and follow one of the hyperlinks of the current webpage. According to [3] the original PageRank algorithm used $\alpha = 0.15$. The PageRank then simulates this diffusion process until it converges to a stationary distribution. The Google rank (PageRank) $G(i)$ of a node i is proportional to the number of random walkers at this node in such steady state. We chose to normalize it so that $\sum_i G(i) = 1$ but in general the normalization factor does not matter as ranking relies on relative values of $G(i)$ for different webpages. When one enters a search keyword such as e.g. "statistical physics" on the Google website the search engine first localizes the subset of webpages containing this keyword and then simply presents them in the descending order based on their PageRank values. The main equation determining the PageRank values $G(i)$ for all webpages in the WWW is

$$G(i) = \alpha + \sum_{j\to i}(1-\alpha)\frac{G(j)}{K_{out}(j)}. \tag{1}$$

Here $K_{out}(j)$ denotes the the number of hyperlinks (the out-degree) in the node j and the summation goes over all nodes j that have a hyperlink pointing to the node i. In the matrix formalism the PageRank values are given by the components of the principal eigenvector of an asymmetric positive matrix related to the adjacency matrix of the network. Such eigenvector could be easily found using a simple iterative algorithm [3]. The fast convergence of this algorithm is ensured by the fact that the adjacency matrix of the network is sparse. We first consider the effect of the community structure on Google ranking in the simplest and most physically transparent case of $\alpha = 0$. In order for the algorithm to properly converge in this case we need to assume that $K_{out}(i) > 0$ for all nodes in the network. Consider a network in which N_c nodes form a community characterized by higher than average density of edges linking these nodes to each other. Let E_{cw} denote the total number of hyperlinks pointing from nodes in the *community* to the outside *world*, while E_{wc} - the total number of hyperlinks pointing in the opposite direction (See Fig. 1 for an illustration). Similarly E_{cc} and E_{ww} denote the total number of links connecting nodes within the community and, respectively, the outside world. The total number of hyperlinks pointing to nodes inside the community is given by $E_{cc} + E_{wc} = N_c\langle K_{in}\rangle_c$ where $\langle K_{in}\rangle_c$ is the average in-degree of community nodes. Similarly, $E_{cc}+E_{cw} = N_c\langle K_{out}\rangle_c$, where $\langle K_{out}\rangle_c$ is the average out-degree in the community, gives the total number of hyperlinks originating on community nodes. The Google rank is computed in the steady state of the diffusion process where the average number of random surfers currently visiting any given webpage does not change with time. This means that the total current of surfers J_{cw} leaving the community for the outside world must be precisely balanced by the current J_{wc} entering the community during the same time interval. Let $G_c = \langle G(i)\rangle_{i\in C}$ denote the average Google rank inside the community given by the average number of random surfers on its nodes. If edges pointing away from the community to the outside world start at an unbiased selection of nodes in the community the average current flowing along any of those edges would be given by $G_c/\langle K_{out}\rangle_c$ while the total current leaving the community $J_{cw} = E_{cw}G_c/\langle K_{out}\rangle_c$. Similar analysis gives $J_{wc} = E_{wc}G_w/\langle K_{out}\rangle_w$, where $\langle K_{out}\rangle_w$ is the average out-degrees of nodes in the world outside the community. Balancing these two currents one gets:

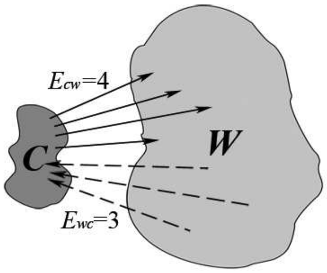

FIG. 1: The illustration of hyperlink connections between the community C and the outside world W. E_{cw} and E_{wc} are numbers of links from the community to the outside world and from the outside world to the community, respectively.

$$\frac{G_c}{G_w} = \frac{E_{wc}}{E_{cw}}\cdot\frac{\langle K_{out}\rangle_c}{\langle K_{out}\rangle_w} \quad . \tag{2}$$

The Eq. 2 is based on the "mean-field" assumption that average values of the Google rank and the out-degree on those community nodes that actually send links to the outside world are equal to their overall average values inside the community [6]. It is tempting to assume that higher than average density of hyperlinks connecting nodes in the community is beneficial for the Google rank of its nodes as it "traps" random surfers to spend more time within the community. It turned out that this naive argument is not necessarily true. In fact one is equally likely to observe an opposite effect: an excess of intra-community links could lead to a lower than average Google rank of its nodes. To see it explicitly one should replace E_{wc} and E_{cw} in Eq. 2 with identical expressions $\langle K_{in}\rangle_c N_c - E_{cc}$ and $\langle K_{out}\rangle_c N_c - E_{cc}$ respectively:

$$\frac{G_c}{G_w} = \left(\frac{\langle K_{in}\rangle_c N_c - E_{cc}}{\langle K_{out}\rangle_c N_c - E_{cc}}\right)\cdot\frac{\langle K_{out}\rangle_c}{\langle K_{out}\rangle_w} \quad . \tag{3}$$

From this equation it follows that enhancing the community structure (increasing E_{cc}) while keeping other parameters such as $\langle K_{in}\rangle_c$,$\langle K_{out}\rangle_c$ and $\langle K_{out}\rangle_w$ fixed can

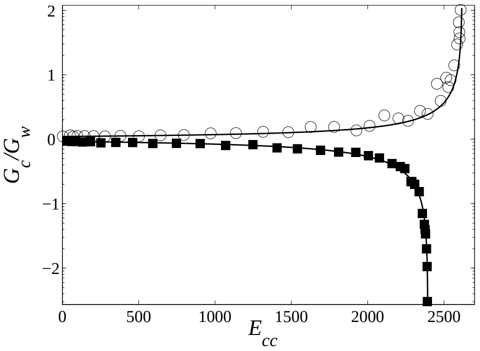

FIG. 2: The ratio of average Google ranks in the community and the outside world G_c/G_w as a function of E_{cc} – the number of intra-community links – in two series of model networks with varying degree of community structure. Open circles correspond to a beneficial effect of the community structure on Google ranking in a scale-free network with $\langle K_{out}\rangle_c = 5.24 < \langle K_{in}\rangle_c = 5.9$. On the other hand, filled squares show a detrimental effect in another series of networks where $\langle K_{out}\rangle_c = 5.6 > \langle K_{in}\rangle_c = 4.8$. Solid lines are fits with the Eq. 3 with a given set of parameters for each of the networks. All networks with $10,000$ nodes have a community of 500 nodes were generated by the Metropolis rewiring algorithm described later on in the text.

be both good and bad for the average Google rank of the community webpages. It depends on $\langle K_{in}\rangle_c/\langle K_{out}\rangle_c$ – the ratio between average in- and out-degrees of community nodes. If the ratio is less than 1 the increase in E_{cc} leads to a further decrease of G_c/G_w below one. If the community constitutes just a small fraction of the whole network one could safely assume that G_w remains approximately constant so that the average Google rank of the community, G_c, has to decrease. Similarly if the ratio is larger than 1, G_c grows with the number of inter-community links E_{cc} (see Fig. 2 for an illustration of both cases). The real-life Google algorithm uses a non-zero value of $\alpha \simeq 0.15$. In this case one needs to consider the contribution to currents J_{cw} and J_{wc} due to surfers' random jumps that do not follow the existing hyperlinks. The total number of random walkers residing on the nodes inside the community is G_cN_c and the probability of them to randomly jump to a node in the outside world is $N_w/(N_c + N_w)$. So the contribution to the outgoing current due to such jumps is given by $\alpha G_cN_cN_w/(N_c + N_w)$ which for $N_c \ll N_w$ can be simplified as αG_cN_c. The total outgoing current then can then be written as $J_{cw} = (1 - \alpha)G_cE_{cw}/\langle K_{out}\rangle_c + \alpha G_cN_c$. Similarly the incoming current J_{wc} is given by $(1 - \alpha)G_wE_{wc}/\langle K_{out}\rangle_w + \alpha G_wN_c$. The Eq. 2 remains valid for $\alpha > 0$ if one replaces E_{wc} and E_{cw} with "effective" numbers of edges E^*_{wc} and E^*_{cw} given by

$$E^*_{cw} = E_{cw}(1-\alpha) + N_c\langle K_{out}\rangle_c\alpha \qquad ;$$
$$E^*_{wc} = E_{wc}(1-\alpha) + N_c\langle K_{out}\rangle_w\alpha \qquad . \qquad (4)$$

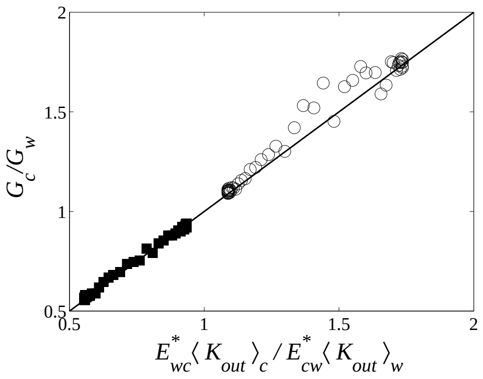

FIG. 3: The ratio of average Google ranks in the community and the outside world G_c/G_w as a function of the ratio of effective numbers of links E^*_{wc}/E^*_{cw}. As predicted by the Eq. 2 these two ratios are basically equal to each other. Different symbols correspond to series of networks described in Fig. 2

These effective numbers take into account contributions to both currents due to random jumps. For a numerical test of the validity of our analytical results we generated an ensemble of directed networks with scale-free distributions of in- and out-degrees: $P(K_{in}) \sim K_{in}^{-2.1}$ and $P(K_{out}) \sim K_{out}^{-2.5}$ correspondingly. The exponents were selected to be identical to their values in the actual WWW network [2, 7]. The community structure in those networks was artificially created using the Metropolis rewiring algorithm described in the next section. As a result a pre-selected group of N_c nodes formed an artificial community with the exact number of intra-community links controlled by the parameters of our simulation. The Fig. 3 shows the results of a numerical test of Eq. 2 in those model networks. For numerical studies of networks with a community structure one needs an efficient algorithm to generate them. In this work we propose a version of the Metropolis random rewiring algorithm introduced earlier in [8]. The algorithm starts from a "seed" network with the desired (scale-free in our case) distributions of in- and out-degrees. Such a seed network can be created e.g. using a stub reconnection procedure described in [9]. The heart of our algorithm is the local rewiring (edge switching) step which strictly conserves separately the in- and out-degrees of every node involved [10]. The only parameters of the Metropolis part of our algorithm are an auxiliary Hamiltonian (energy function) $H = -E_{cc}$ defined as the negative of the number of intra-community links and the inverse temperature β. The steps of the algorithm are as follows: 1) Randomly pick two links, say A→B and C→D; 2) Attempt to rewire them (switch their neighbors) to A→D and C→B. If at least one of these two new links already exists in the network, abort this step and go back to step 1; 3) If the rewiring step decreases the Hamiltonian H it is always accepted, while if it increases the Hamiltonian by ΔH it is accepted only with probability $\exp(-\beta\Delta H)$. If the rewiring step is rejected on steps 2 or 3, the network is returned to the original configuration A→B and C→D;

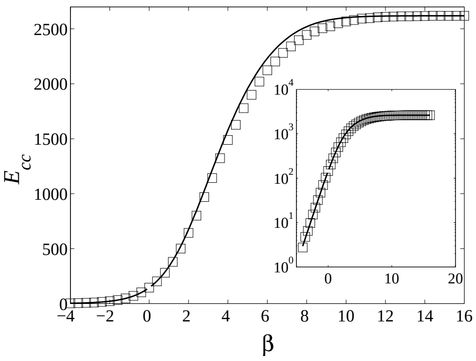

FIG. 4: The number of intra-community links E_{cc} in networks generated by the rewiring algorithm as a function of the inverse temperature β. Negative values of β correspond to networks with anti-community structure and are generated by changing the sign in front of the Hamiltonian H. The solid line is the fit with the analytical expression obtained by solving the Eq. 5 for E_{cc}. The inset shows the same plot with a logarithmic scale of the Y-axis.

4) Repeat the above steps until the number of links inside the community E_{cc} reaches a steady state value. The reciprocal temperature β thus indirectly determines the number of links within the community $E_{cc}(\beta)$ so that an ordinary random (scale-free) network without any community structure is realized at an "infinite temperature" ($\beta = 0$), while the algorithm run at zero temperature ($\beta = \infty$) produces a network with the largest possible number of links within the community. One could also invert the sign in the definition of the Hamiltonian $H = E_{cc}$. Formally this can be thought of as running the algorithm with the original Hamiltonian but a negative inverse temperature $\beta < 0$. Large negative values of β generate networks with an anti-community structure in which the number of intra-community links is lower than in a random network. The relation between E_{cc} and β for both positive and negative values of β is shown in Fig. 4. To analytically derive the relation between E_{cc} and the reciprocal temperature β, we consider the detailed balance in the steady state of the rewiring procedure, in which the probabilities of an increase and a decrease in E_{cc} must be equal to each other. E_{cc} is increased by 1 if the links picked at a given step of the rewiring algorithm are C→W and W→C (here C stands for any node inside the community and W - in the outside world). The probability to pick such pair is proportional to $E_{cw}E_{wc}$. On the other hand, if the selected links are C→C and W→W the number of links in the community would decrease by one with a probability $\exp(-\beta)$. All other selections of links do not change the E_{cc}. The detailed balance equation for the rewiring procedure thus reads:

$$E_{cw}E_{wc} = E_{cc}E_{ww}e^{-\beta} \qquad (5)$$

Additional constraints (i) $E_{cc} + E_{wc} = \langle K_{in}\rangle_c N_c$ (the sum of in-degrees of all nodes within the community), (ii) $E_{cc} + E_{cw} = \langle K_{out}\rangle_c N_c$ (the sum of out-degrees of all nodes within the community) and (iii) $E_{cc} + E_{cw} + E_{wc} = E$ (the total number of edges in the network) plugged into the Eq. (5) result in a quadratic equation for E_{cc} as a function of $\langle K_{in}\rangle_c$, $\langle K_{out}\rangle_c$, E, and β – the parameters strictly conserved in our rewiring algorithm. The Fig. 4 compares the analytical expression for $E_{cc}(\beta)$ obtained by solving the Eq. 5 with numerical simulations for different values of β. Clearly, E_{cc} increases with β in general accord with the Eq. 5. When β is sufficiently large, E_{cc} exponentially approaches a limiting value equal to $\max(\langle K_{in}\rangle_c, \langle K_{out}\rangle_c)N_c$ – the maximal number of links within a community given the set of in- and out-degrees of its nodes. The deviations between the analytical formula and numerical results visible for large values of β could be attributed to the "no multiple edges" restriction in networks generated by our rewiring algorithm. As the density of inter-community links increases with β more and more of the rewiring steps leading to an increase of E_{cc} have to be aborted as the new link they are attempting to create within a community already exists. This situation is more appropriately described by the following equation: $E_{cw}E_{wc}(1 - E_{cc}/E)(1 - E_{ww}/E) = E_{cc}E_{ww}(1 - E_{cw}/E)(1 - E_{wc}/E)e^{-\beta}$, reminiscent of the detailed balance equation in two-fermion scattering (see also [11]).

In summary, we investigated how the WWW community structure affects the Google rank of webpages belonging to a given community. We have shown that depending on the balance between average in- and out-degrees of webpages inside the community the excess density of intra-community hyperlinks can either boost or decrease the average Google ranking of its webpages. For numerical studies of scale-free networks with a community structure we developed a version of the Metropolis rewiring algorithm first proposed by one of us in [8]. This algorithm allows one to generate a random network with a desired density of intra-community links and a given distribution of in- and out-degrees.

Work at Brookhaven National Laboratory was carried out under Contract No. DE-AC02-98CH10886, Division of Material Science, U.S. Department of Energy.

[1] S. Brin and L. Page, Computer Networks and ISDN Systems, **30**, 107 (1998).

[2] R. Kumar, P. Raghavan, S. Rajagopalan, and A.Tomkins, Computer Networks **31**, 11 (1999).

[3] L. Page, S. Brin, R. Motwani and T. Winograd, Stanford Digital Library Technologies Project (1998).

[4] S. Bilke and C. Peterson Physical Review E **64**, 036106 (2001).

[5] K. A. Eriksen, I. Simonsen, S. Maslov and K. Sneppen, Phys. Rev. Lett. **90**, 148701 (2003).

[6] Strictly speaking the selection of nodes sending links to the outside world is biased to those with higher values of K_{out} and thus might be not representative of the whole community. However, generally speaking this does not contradict our mean-field approximation as (at least in random scale-free networks) there is no correlation between the out-degree of a node and its Google ranking.

[7] A. Albert, H. Jeong, and A.-L. Barabsi, Nature, **401**, 130, (1999).

[8] S. Maslov, K. Sneppen, A. Zaliznyak, Physica A, **333**, 529 (2004).

[9] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Phys. Rev. E **64**, 161 (1995).

[10] S. Maslov and K. Sneppen, Science, **296**, 910 (2002).

[11] J. Park and M. E. J. Newman, Phys. Rev. E **68**, 026112 (2003).